May 1993

A Kirchhoff approach to seismic modeling and prestack depth migration

Zhenyue Liu

Partially supported by the Office of Naval Research,
Contract Number N00014-91-J-1267

Center for Wave Phenomena
Colorado School of Mines
Golden, Colorado 80401
303/273-3557

A Kirchhoff approach to seismic modeling and prestack depth migration

Zhenyue Liu

ABSTRACT

The Kirchhoff integral provides a robust method for implementing seismic modeling and prestack depth migration, which can handle lateral velocity variation and turning waves. With a little extra computation cost, the Kirchhoff-type migration can obtain multiple outputs that have the same phase but different amplitudes, compared with that of other migration methods. The ratio of these amplitudes is helpful in computing some quantities such as reflection angle.

Here, I develop a seismic modeling and prestack depth migration method based on the Kirchhoff integral, that handles both laterally variant velocity and a dip beyond 90 degrees. The method uses a finite-difference algorithm to calculate traveltimes and WKBJ amplitudes for the Kirchhoff integral. Compared to ray-tracing algorithms, the finite-difference algorithm gives an efficient implementation and single-valued quantities (first arrivals) on output. In my finite-difference algorithm, the upwind scheme is used to calculate traveltimes, and the Crank-Nicolson scheme is used to calculate amplitudes. Moreover, interpolation is applied to save computation cost.

The modeling and migration algorithms here require a smooth velocity function. I develop a velocity-smoothing technique based on damped least-squares to aid in obtaining a successful migration. This velocity-smoothing technique also can be used to improve results of other migration algorithms, such as Gaussian beam migration.

INTRODUCTION

Seismic modeling and migration play an important role in seismic data processing. To treat complex media, one needs prestack depth migration that can handle lateral velocity variation and reflector dips. Conventional techniques, such as the downward continuation of sources and geophones by finite difference (S-G finite-difference migration), are relatively slow and dip-limited. Compared to S-G finite-difference migration, the Kirchhoff integral implements prestack migration relatively efficiently and has no dip limitation.

Bleistein et al. (1987) derived the Kirchhoff integral method by using the WKBJ approximation. This method treats amplitude in migration in a WKBJ-consistent manner so that the output is the reflectivity function. Furthermore, Bleistein showed that one can design weights in the Kirchhoff integral to determine quantities such as the reflection angle, based on the stationary-phase principle. In Bleistein's integral formula, the integrand consists of traveltime, WKBJ amplitude, ray parameters, and so on. Two commonly used approaches to calculate traveltimes and the other quantities in the Kirchhoff integral are ray-tracing and finite-difference applied to the eikonal equation. The ray-tracing algorithm calculates traveltimes and amplitudes on each ray. Usually, the rays do not pass through output grid points, so we need to interpolate to obtain traveltimes and amplitudes at the grid locations. This interpolation is complicated when the raypaths have a caustic, and thus limits the efficiency of the ray-tracing algorithm. *In contrast, the finite-difference algorithm* calculates traveltimes and amplitudes directly at output grid locations. For caustics, the finite-difference algorithm calculates the first-arrival traveltime and the corresponding amplitudes.

Traveltimes satisfy the eikonal equation, and amplitude terms satisfy linear partial differential equations that depend on traveltime derivatives. Van Trier and Symes (1990) introduced a finite-difference scheme for solving this equation to obtain the traveltimes and traveltime derivatives. Pusey and Vidale (1991) used a similar, explicit scheme to solve for the WKBJ amplitudes. For the explicit scheme, one has to choose a small enough step size for a stability when the velocity function is varied. *In* this paper, I use the Crank-Nicolson scheme to solve for the amplitudes. Compared to the explicit scheme, the Crank-Nicolson scheme is second-order accurate and absolutely stable so that computation cost is made relatively small for variable velocity by choosing large step sizes.

Calculation cost of pointwise traveltimes and amplitudes by the above method would dominant the total calculation cost of the Kirchhoff integral method. For each source or receiver, the calculation cost depends on the number of grid points. To speed up the Kirchhoff integral method, I apply interpolation to avoid computation on the entire grid.

A smooth velocity function is required by most migration algorithms. A few methods for smoothing the model velocity have been developed in the past. The windowed-averaging method, for example, calculates a smooth velocity value at one point by averaging velocity values over a depth-midpoint window centered at this point. If the original velocity is discontinuous, it is easy to show that the smoothed velocity from the windowed-averaging method does not have the continuity of the first derivative. Consequently, this method is not effective when the velocity has a strong discontinuity. Here, I present a velocity-smoothing technique based on damped least-squares. The curvature of the raypath is a key factor in migration imaging. The smaller the curvature, the more stable the migration result. I derive a representation for the curvature that depends on first derivatives of velocity. A smooth velocity

function is sought that minimizes the weighted sum of (1) the deviation between the smooth velocity and the original one, and (2) the first derivatives of velocity with respect to spatial variables. I use the result that the reciprocal of the curvature of the raypath is proportional to the velocity gradient. Therefore, my method may remove migration artifacts by suppressing the curvature of the raypath. Moreover, this method allows local variation in the degree of smoothing allowed by using a window function, so that the velocity is smoothed more in rougher areas.

FINITE-DIFFERENCE ALGORITHM

The Kirchhoff integral method can be represented by

$$output = integral\{weight \cdot input\}.$$

For modeling problems, the input consists of velocity layers, and the output is seismic traces; for inversion problems, the input is seismic traces, the output is a structural image. Both require common quantities to be calculated. These quantities include (Bleistein, 1986):

τ	traveltime,
ϕ	propagation angle,
σ	running ray parameter,
β	incident angle from source or reciever,
$\partial\beta/\partial x$	geometrical spreading parameter.

For each source or receiver, the above quantities satisfy the follow eqations:

$$\left(\frac{\partial\tau}{\partial x}\right)^2 + \left(\frac{\partial\tau}{\partial z}\right)^2 = \frac{1}{v^2(x,z)}, \tag{1}$$

$$\sin\phi = v\frac{\partial\tau}{\partial x}, \tag{2}$$

$$\frac{\partial\sigma}{\partial x}\frac{\partial\tau}{\partial x} + \frac{\partial\sigma}{\partial z}\frac{\partial\tau}{\partial z} = 1, \tag{3}$$

$$\frac{\partial\beta}{\partial x}\frac{\partial\tau}{\partial x} + \frac{\partial\beta}{\partial z}\frac{\partial\tau}{\partial z} = 0, \tag{4}$$

$$\frac{\partial}{\partial z}\left(\frac{\partial\beta}{\partial x}\right) + \frac{\partial}{\partial x}\left[\mu(x,z)\frac{\partial\beta}{\partial x}\right] = 0, \tag{5}$$

where $v(x,z)$ is velocity and

$$\mu(x,z) = \frac{\partial\tau}{\partial x}\left[\frac{\partial\tau}{\partial z}\right]^{-1}.$$

Equation (1) is the eikonal equation. Equations (3) and (4) are derived by Pusey and Vidale (1991). Equation (5) follows from equation (3). Van Trier et al. (1990) used the efficient *upwind* scheme to solve this equation for traveltime and traveltime derivatives. However, computation in Cartesian coordinates requires that $\partial\tau/\partial z$ be positive, so turning waves cannot be handled. Computation in polar coordinates does not suffer this limitation. Let us introduce the coordinate transform

$$x = x_s + r\sin\theta, \qquad z = r\cos\theta.$$

In polar-coordinates, equations (1) through (5) become

$$\left(\frac{\partial\tau}{\partial r}\right)^2 + \frac{1}{r^2}\left(\frac{\partial\tau}{\partial\theta}\right)^2 = \frac{1}{v^2(r,\theta)}, \tag{6}$$

$$\sin(\phi-\theta) = \frac{v}{r}\frac{\partial\tau}{\partial\theta}, \tag{7}$$

$$\frac{\partial\sigma}{\partial r}\frac{\partial\tau}{\partial r} + \frac{1}{r^2}\frac{\partial\sigma}{\partial\theta}\frac{\partial\tau}{\partial\theta} = 1, \tag{8}$$

$$\frac{\partial\beta}{\partial r}\frac{\partial\tau}{\partial r} + \frac{1}{r^2}\frac{\partial\beta}{\partial\theta}\frac{\partial\tau}{\partial\theta} = 0, \tag{9}$$

$$\frac{\partial}{\partial r}\left(\frac{\partial\beta}{\partial\theta}\right) + \frac{\partial}{\partial\theta}\left[\mu(r,\theta)\frac{\partial\beta}{\partial\theta}\right] = 0, \tag{10}$$

where

$$\mu(r,\theta) = \frac{\partial\tau}{\partial\theta}\left[r^2\frac{\partial\tau}{\partial r}\right]^{-1}.$$

Although more computation is required in polar-coordinates, the boundary conditions are more easily treated and, more importantly, we can handle turning waves.

The coefficient μ of equation (10) is inside the differential operator. If an error exists in numerical computation of μ, the solution of equation (10) may suffer. Instead of solving equation (10), I use an approximate formula

$$\frac{\partial\beta}{\partial\theta} = \frac{v_0 r}{\sigma}\cos(\theta-\phi), \tag{11}$$

where v_0 is the velocity value at the source position. Formula (11), derived in Appendix A, is exact for linear velocity. When velocity is strongly nonlinear, the geometry-spreading factor based on formula (11) is not accurate. However, not only the geometry-spreading factor but also the finite difference algorithm, which requires that $\partial\tau/\partial r$ be positive, suffers from strong-nonlinear velocity . The smoothing technique in this paper will produce a sufficiently smooth velocity to solve these two problems.

Crank-Nicolson scheme

Van Trier et al. (1990) gives an upwind finite-difference scheme for solving equa tion (6). Also that scheme yields derivative values for use in equation (7). Pusey and Vidale (1991) use a similar scheme to solve equation (8). Those upwind schemes are explicit. Therefore, one has to choose a small enough step size for stability when the velocity function is varied. In this paper, the Crank-Nicolson scheme is applied to equations (8) and (9). Compared to the explicit scheme, the Crank Nicolson scheme is second-order accurate and absolutely stable so that computation cost is made relatively small for variable velocity by choosing large steps.

Let

$$p = \frac{\partial \tau}{\partial r}, \qquad q = \frac{1}{r^2}\frac{\partial \tau}{\partial \theta}.$$

With the Crank-Nicolson approximation, equation (8) becomes

$$\frac{\sigma_i^{j+1} - \sigma_i^j}{2\Delta r}(p_i^{j+1} + p_i^j) + \frac{q_i^{j+1} + q_i^j}{8\Delta\theta}(\sigma_{i+1}^{j+1} - \sigma_{i-1}^{j+1} + \sigma_{i+1}^j - \sigma_{i-1}^j) = 1, \qquad (12)$$

and equation (9) becomes

$$\frac{\beta_i^{j+1} - \beta_i^j}{2\Delta r}(p_i^{j+1} + p_i^j) + \frac{q_i^{j+1} + q_i^j}{8\Delta\theta}(\beta_{i+1}^{j+1} - \beta_{i-1}^{j+1} + \beta_{i+1}^j - \beta_{i-1}^j) = 0, \qquad (13)$$

where j is the index of r, i is the index of θ, and Δr and $\Delta\theta$ define the grid cell sizes.

For each j, equations (12) and (13) are tridiagonal systems that can be solved relatively efficiently.

INTERPOLATION TECHNIQUE

Calculation cost of pointwise traveltimes and amplitudes by the above method would dominant the total calculation cost of the Kirchhoff integral method. For each source or receiver, the calculation cost depends on the number of grid points. To speed up the Kirchhoff integral method, I apply interpolation at two stages so as to avoid computation on the entire grid.

One stage is the interpolation between the sources or receivers: I calculate traveltimes and amplitudes only at selected sources or receivers, and then interpolate traveltimes and amplitudes at the other sources or receivers. Suppose that the traveltime functions $\tau(x, z; x_{s_1})$ and $\tau(x, z; x_{s_2})$ from two sources x_{s_1} and x_{s_2} have been calculated. I interpolate a traveltime function from source x_s ($x_{s_1} < x_s < x_{s_2}$) by

$$\tau(x, z, x_s) = \lambda\tau(x + x_{s_2} - x_s, z; x_{s_2}) + (1 - \lambda)\tau(x + x_{s_1} - x_s, z; x_{s_1}), \qquad (14)$$

where $\lambda = (x_s - x_{s_1})/(x_{s_2} - x_{s_1})$. When the velocity is only depth-dependent, this linear interpolation is exact because the traveltime is invariant for parallel movement; i.e., for any h,

$$\tau(x + h, z; x_s + h) = \tau(x, z; x_s).$$

For a general velocity function $v(x,z)$, the interpolation error depends on $\partial v/\partial x$. Reducing this error is one motivation for smoothing the velocity.

The other interpolation is between grid points. To save computation, I use a relatively coarse grid in the traveltime and amplitude calculation. Then, to preserve resolution, I use a finer grid for migration output with traveltimes and amplitudes calculated by bilinear interpolation.

VELOCITY SMOOTHING

A smooth velocity function is required by most migration algorithms. Before a smoothing technique is selected, it is essential to know how smoothness of velocity affects migration results.

Numerous migration algorithms are based on ray theory (high-frequency assumption). The smaller the curvature or the larger the radius of curvature, the better is the quality of the raypath. Generally, a large curvature increases the sensitivity of a raypath to the incident angle from the source, resulting in sparse raypath coverage. In addition, a large curvature may cause troubles in the imaging process. In Gaussian beam migration, for example, traveltime at one point in the vicinity of a central ray are calculated by projecting this point onto the ray and then by using the Taylor series expansion. If the curvature of the ray is too large, the projecting point is not unique and also the Taylor series expansion is inaccurate, so that a poor migration result may suffer from the inaccurate calculation of the traveltimes. For another example, the finite-difference method in this paper will fail if, at any point of the raypath, there is more than a 90-degree difference between the ray direction and the direction from the source position to this point. Therefore, control of the curvature of the raypath is necessary. From Dohr (1985, p. 23), the curvature κ can be represented by

$$\kappa \equiv \frac{1}{R} = -\frac{\partial v}{\partial x}\frac{\cos\theta}{v} + \frac{\partial v}{\partial z}\frac{\sin\theta}{v}, \tag{15}$$

where R is the radius of the curvature, v is the velocity, and θ is the propagation angle. Equation (15) shows that the first derivatives dominate the curvature of the ray. Based on this characteristic, I used the damped least-squares method to calculate a smooth velocity that has suppressed the first derivatives and, therefore, reduced the raypath curvature. Use of a smooth velocity function is also required in the WKBJ approximation and in the interpolation for traveltimes.

Let $v(x,z)$ be the original velocity; then a smooth velocity $v_s(x,z)$ is determined by

$$\int (\bar{v}_s(x,z) - v(x,z))^2\,dx + \alpha_x^2 \int w(x,z)\left(\frac{\partial \bar{v}_s}{\partial x}\right)^2 dx = \min, \tag{16}$$

$$\int (v_s(x,z) - \bar{v}_s(x,z))^2\,dz + \alpha_z^2 \int w(x,z)\left(\frac{\partial v_s}{\partial z}\right)^2 dz = \min, \tag{17}$$

where α_x, α_z are the *smoothing parameters* and $w(x,z)$ is a window function. The window function, ranging from 0 to 1, allows local variation in the degree of smoothing desired. Roughness of the original velocity function is usually not uniform, so one can design a window function such that velocity is smoothed more in rougher areas.

Equation (16) smooths the original velocity along the x-direction, from which $\bar{v}_s(x,z)$ is obtained; equation (17) smooths $\bar{v}_s(x,z)$ along the z-direction to get the smoothed velocity $v_s(x,z)$. In the wavenumber domain and for $w(x,z) \equiv 1$, the smooth velocity can be determined by

$$V_s(k_x,k_z) = \frac{V(k_x,k_z)}{(1+\alpha_x^2 k_x^2)(1+\alpha_z^2 k_z^2)}, \tag{18}$$

where V_s and V are the Fourier transforms of v_s and v respectively. Therefore, large-wavenumber components are suppressed. In discretization, equation (18) becomes

$$V_s(k_x,k_z) = \frac{V(k_x,k_z)}{(1+4(\alpha_x/\Delta x)^2\sin^2(k_x\Delta x/2))\,(1+4(\alpha_z/\Delta z)^2\sin^2(k_z\Delta z/2))}, \tag{19}$$

where Δx and Δz define the grid-cell sizes. A proof of formula (18) and formula (19) is in Appendix B. I define the unit of the smoothing parameters by

$$\alpha_x = \Delta x/2, \qquad \alpha_z = \Delta z/2.$$

For such smothing parameters, the Nyquist-wavenumber components in x and z directions, respectively, reduce by 1/2, i.e.,

$$V_s(2/\Delta x, 2/\Delta z) = \frac{1}{4} V(2/\Delta x, 2/\Delta z).$$

The larger the values of α_x and α_z, the smoother will be $v_s(x,z)$ and the larger will be the difference between $v(x,z)$ and $v_s(x,z)$. The following formula may be used to measure the difference between $v(x,z)$ and $v_s(x,z)$:

$$\epsilon^2(z) = \frac{\int\int_0^z (v_s(x,z') - v(x,z'))^2 dz'dx}{\int\int_0^z v(x,z')^2 dz'dx}. \tag{20}$$

Based on my experience,if $\epsilon(z)$ is greater than 0.1, the smoothing parameters are too large—the velocity is oversmoothed.

COMPUTER IMPLEMENTATION

I applied the modeling and migration method to synthetic models. Also, I tested the effect of velocity smoothing on the Kirchhoff migration of this paper and Gaussian beam migration.

Synthetic Data

The first model shown in Figure 1a, consists of five reflectors, each with a dipping and horizontal segment. Dips for the dipping segments range from 30 to 90 degrees in 15-degree increments. The velocity function consists of a linear function, $v(z) = 1.5+z$ km/s, plus a lateral variation that is $\cos(\frac{\pi}{2}(x-1.5))$ multiplied by a depth-dependent function, as shown in Figure 1b. The maximum lateral variation is 12 percent at $z = 1$ km. With the model and the velocity function, I generated two data sets: zero-offset and an offset of 1000 m. Figures 2 and 3 show the modeling data and inversion results for the two offsets. Because of the lateral variation in velocity, even horizontal reflectors give curve events in the data. After migration, all reflectors, horizontal through vertical, are imaged to their correct positions, for both the zero- and nonzero-offset data. Figure 4 shows Gaussian beam migration result by using the same zero-offset data. While both our method and Gaussian beam migration did well, the Gaussian beam migration used 17 minutes on the IBM Risk System/6000, and our method used 7 minutes, even though I used the prestack algorithm for the zero-offset data.

The second example is on turning-wave migration. The model shown in Figure 5 is a single reflector, with a segment beyond 90 degrees. The velocity function is the same as in the first example, shown in Figure 1b. With the model and the velocity function, I generated a data set with an offset of 500 m, shown in Figure 6a. After migration, the reflector is imaged to its correct position shown in Figure 6b, even for the segment beyound 90-degrees. This result shows that my method can handle lateral variation in velocity and turning waves.

Smoothing test

The velocity model shown in Figure 7 consists of six constant-velocity layers and the input zero-offset data for migration shown in Figure 8 were generated with this model by the CSHOT program(Docherty, 1987).

Different smoothed velocity functions are used in Gaussian-beam migration that is implemented by Hale's program (1992). Figure 9 shows the data migrated with the unsmoothed velocity in Figure 7. The migration result is poor: low resolution and migration artifacts are in the first three reflectors, and the last two reflectors are almost invisible. Figure 10 shows the data migrated with a smoothed velocity (smoothing parameters $\alpha_x = \alpha_z = 2.5$). While it is much better than Figure 9, there are still some artifacts. Figure 11, showing the data migrated with a more smoothed velocity (smoothing parameter $\alpha_x = \alpha_z = 5$), gives an improved structural image. If too large smoothing parameters ($\alpha_x = \alpha_z = 20$) are used, the migration result, shown in Figure 12, deteriorates. One can see that the corners of the third reflector have become too sharp, and the bottom reflector is no longer flat. Figure 13 shows the data migrated with the smoothed velocity by the windowed averaging method. The window size is selected so that the difference between the smoothed velocity and the original one is about the same as that in Figure 11. Both the damped least-squares

and the windowed averaging method improve structural images through smoothing the velocity, but the former looks a little better than the latter. There is incoherence on the vicinity of the right intersection between the first and the second interface in Figure 13 compared to Figure 11. The selected parts are shown in Figure 14. From this result, one may conclude that if the differences between the smoothed velocities and the original one are the same, the damped least-squares method smooths velocity a little more effectively, therefore, gives a slightly better result.

The velocity in Figure 7 has strong lateral variation. For this unsmoothed velocity, the finite difference algorithm for the eikonal equation 6 will break down because $\partial\tau/\partial z$ may not positive. Therefore, in order to implement the Kirchhoff migration of this paper, the smoothed velocity is required. The least smoothing parameters for this requirement are $\alpha_x = \alpha_z = 14$. Since the velocity is slightly oversmoothed, the bottom reflector is not very flat. Furthermore, like other Kirchhoff migration methods, the discontinuities in the bottom event of Figure 8 make strong diffraction smiles in Figure 10 since large apertures are used. In contrast, Gaussian beam migration does not suffer this problem.

Figure 16 shows the differences between the smoothed velocities and the original one, which is calculated by using formula (20). The larger the smoothing parameters, the larger the difference. This difference may help to choose suitable smoothing parameters. For example, the difference for $\alpha_x = \alpha_z = 20$ is greater than 0.1 from which I conclude that the velocity is oversmoothed and then reject these smoothing parameters.

CONCLUSION

The Kirchhoff integral provides a powerful tool for modeling and migration. In this paper, a finite-difference algorithm is used to calculate traveltimes and amplitudes. With the help of interpolation, this method can be efficiently implemented. The result on the synthetic data shows that this method can handle lateral variation in velocity and turning waves. One limitation is that the Kirchhoff method cannot efficiently deal with caustics in the Green's functions of the integration operator and multiple-arrival times. Another is that this method needs a sufficiently smooth velocity function. When velocity has a strong lateral variation, the eikonal equation cannot be solved by this method. This difficulty may be solved through using smoothed velocity.

ACKNOWLEDGMENTS

Thanks to Dr. Ken Larner for his critical reading of the first drafts of this paper. Thanks to Dr. Norman Bleistein for his help and guidance. The author also gratefully acknowledges the support of the Office of Naval Research, Mathematics Division, and of the members of the Consortium Project on Seismic Inverse Methods for Complex Structures at the Center for Wave Phenomena, Colorado School of Mines.

REFERENCES

Bleistein, N., 1986, Two and one-half dimensional in-plane wave propagation: Geophys. Prosp., **34**, 686-703.

Bleistein, N., Cohen, J., and Hagin, F., 1987, Two and one-half dimensional Born inversion with an arbitrary reference: Geophys., **52**, 26-36.

Docherty, P. C., 1987, Ray theoretical modeling, migration, and inversion in the two-and-one-half-dimensional layered acoustic media: Center for wave phenomena Research Report, CWP-051.

Dohr, G., 1985, Seismic shear waves (part A: theory)· Geophysical Press, London—Amersterdam.

Hale, D, Computational aspects of Gaussian beam migration: CWP Review, 1992.

Pusey, L., and Vidale, J., 1991, Accurate-difference calculation of WKBJ traveltimes and amplitudes: 1991 SEG expanded abstracts, p. 1513.

van Trier, J., and Symes, W., 1990, Upwind finite-difference calculation of seismic traveltimes: 1990 SEG expanded abstracts, p 1000.

APPENDIX A: GEOMETRIC-SPREADING FACTOR FOR LINEAR VELOCITY

Firstly, I assume that the velocity is

$$v(z) = v_0 + az.$$

Following from formulas in Bleistein (1986), I find

$$x - x_s = \frac{v_0}{a \sin\beta}(\cos\beta - \cos\phi), \tag{A-1}$$

and

$$\sigma = \frac{v_0^2(\cos\beta - \cos\phi)}{a \sin^2\beta}. \tag{A-2}$$

From Snell's law

$$\frac{\sin\beta}{v_0} = \frac{\sin\phi}{v(z)},$$

I have

$$\frac{d\phi}{d\beta} = \frac{v(z)\cos\beta}{v_0\cos\phi}. \tag{A-3}$$

By using the Snell's law and equation (A-1), I obtain

$$x - x_s = \frac{v_0}{a}\cot\beta - \frac{v(z)}{a}\cot\phi. \tag{A-4}$$

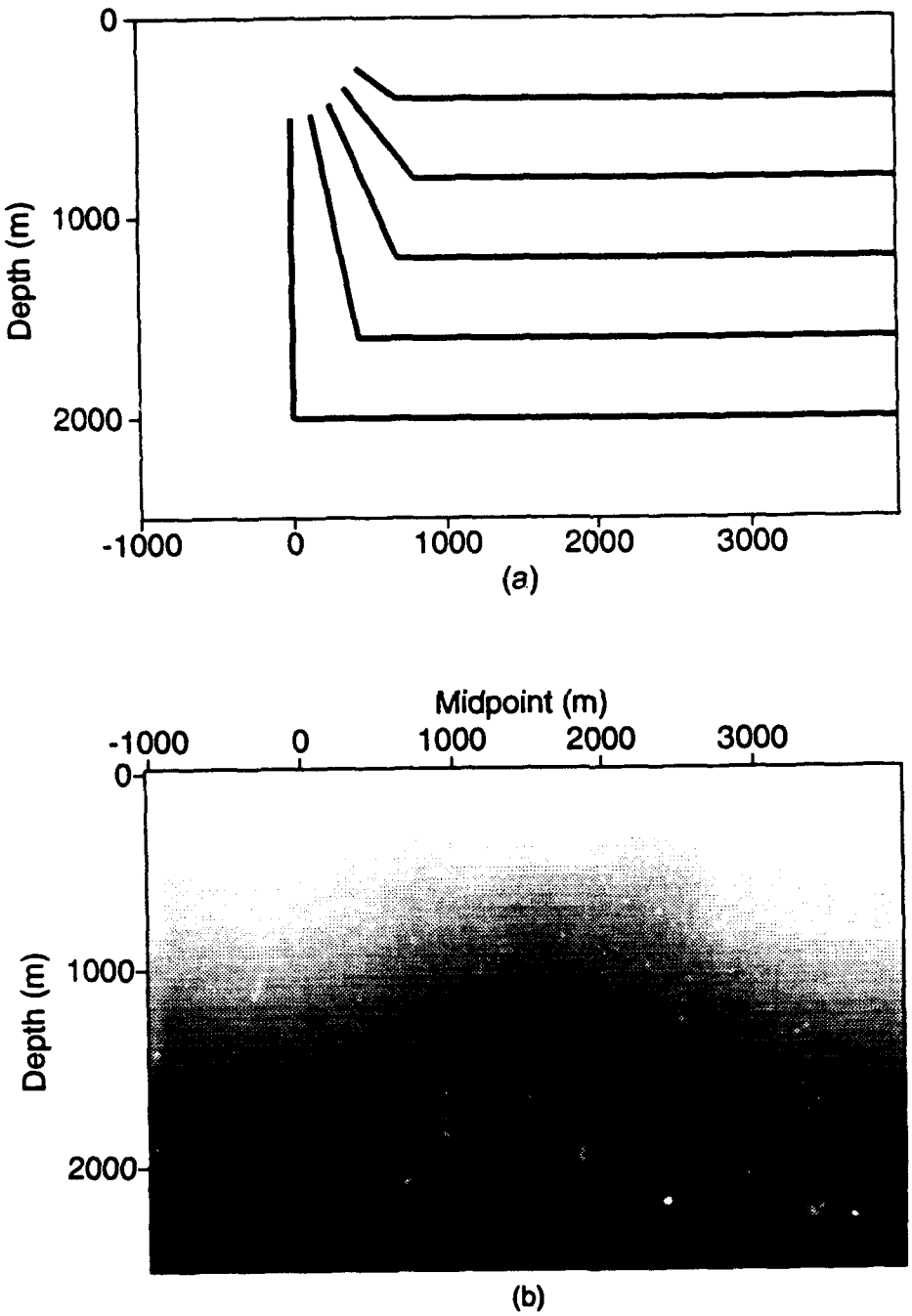

FIG. 1. (a) Subsurface structural model used to generate synthetic seismic traces. (b) Velocity model. The darker shading denotes higher velocity.

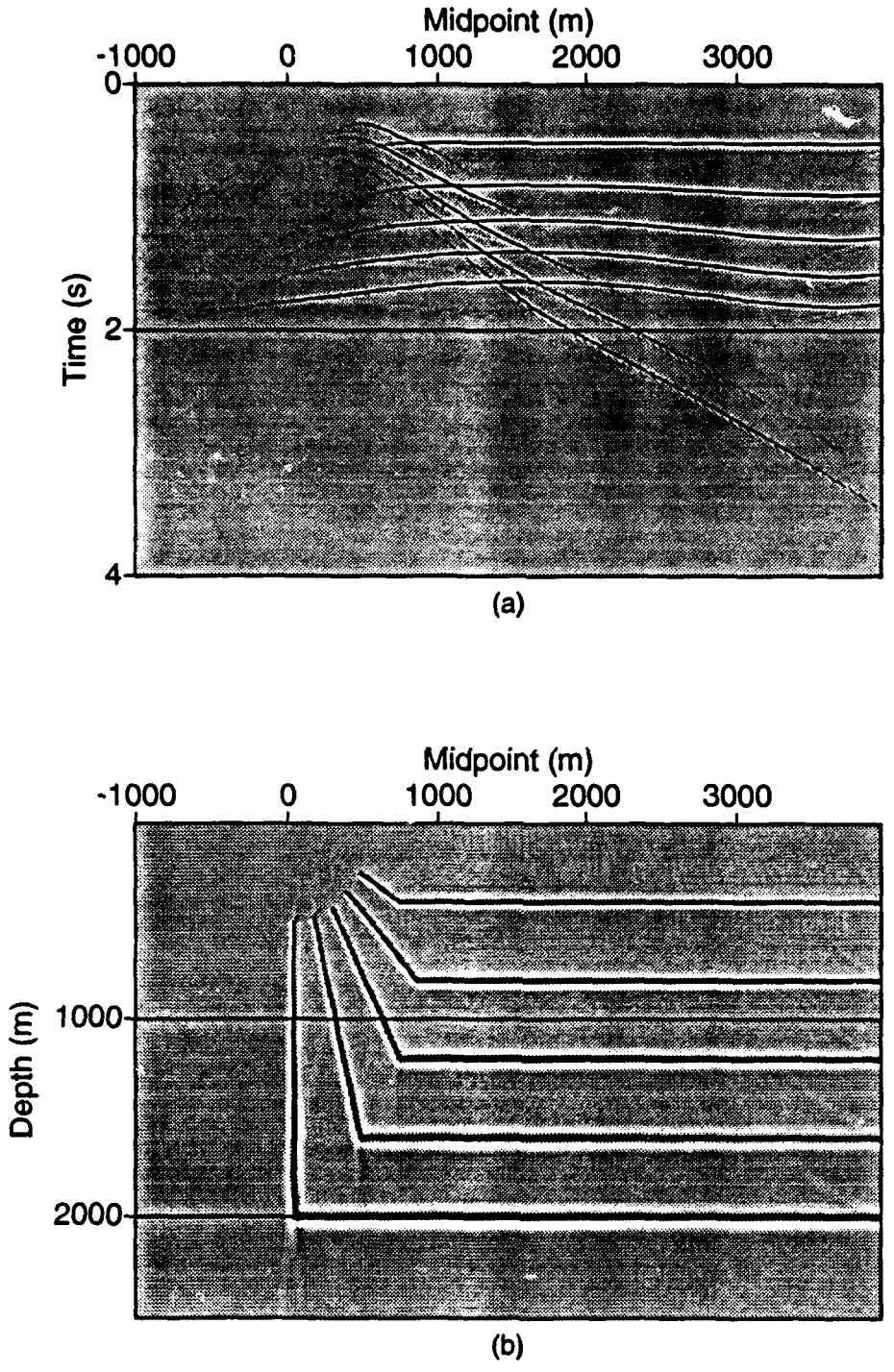

FIG. 2. (a) Zero-offset data for the model in Figure 1. (b) The Kirchhoff migration of the data in (a), with velocity model in Figure 1b.

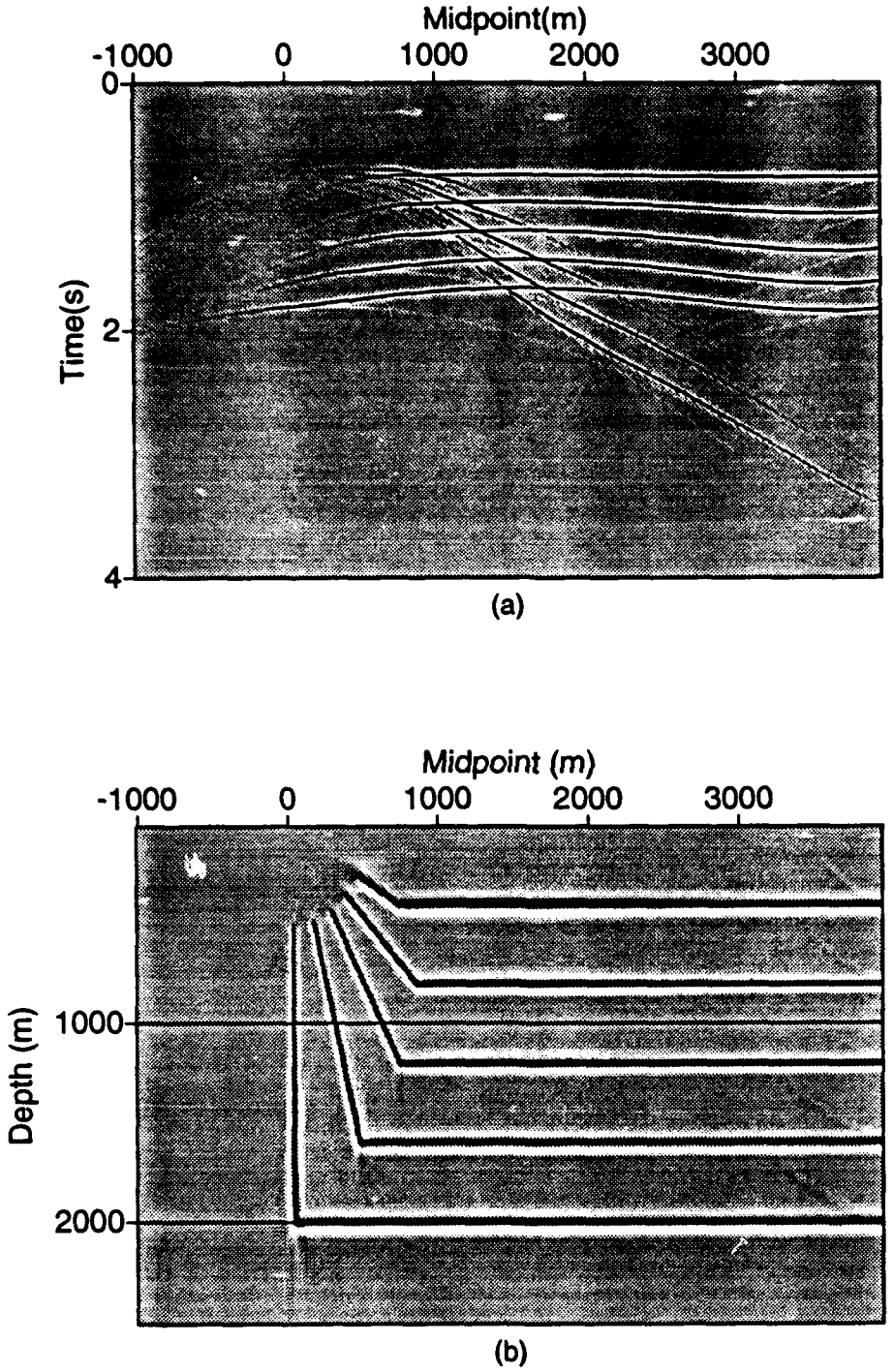

FIG. 3. (a) Nonzero-offset data (offset=1000 m) for the model in Figure 1. (b) The Kirchhoff migration of the data in (a), with velocity model in Figure 1b.

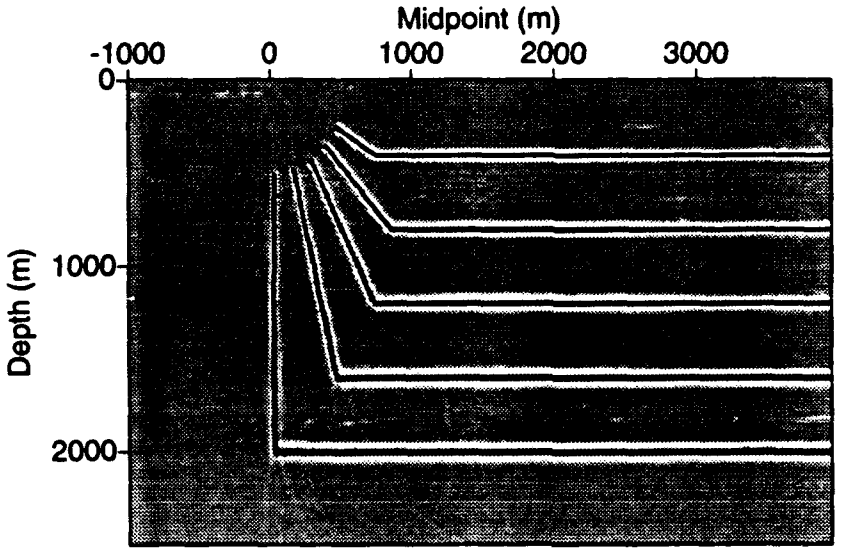

FIG. 4. Gaussian-beam migration of the data in Figure 2a.

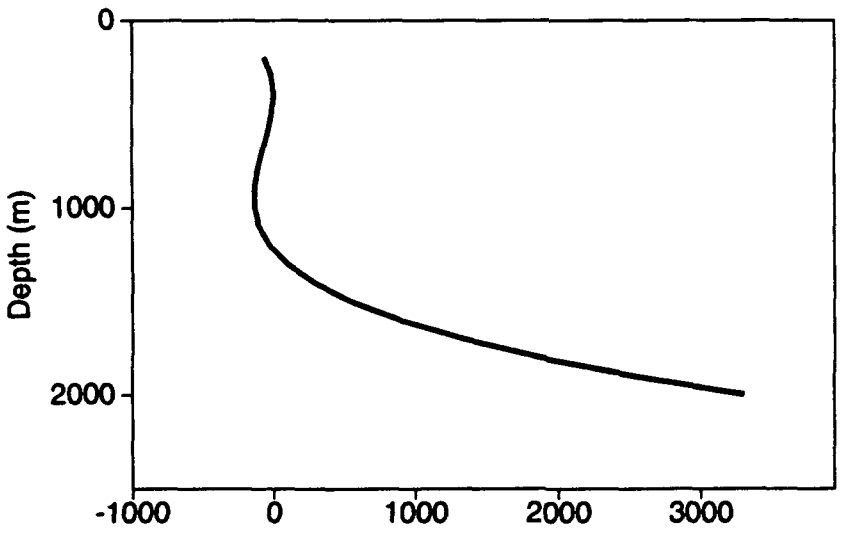

FIG. 5. Subsurface structural model used to generate turning waves.

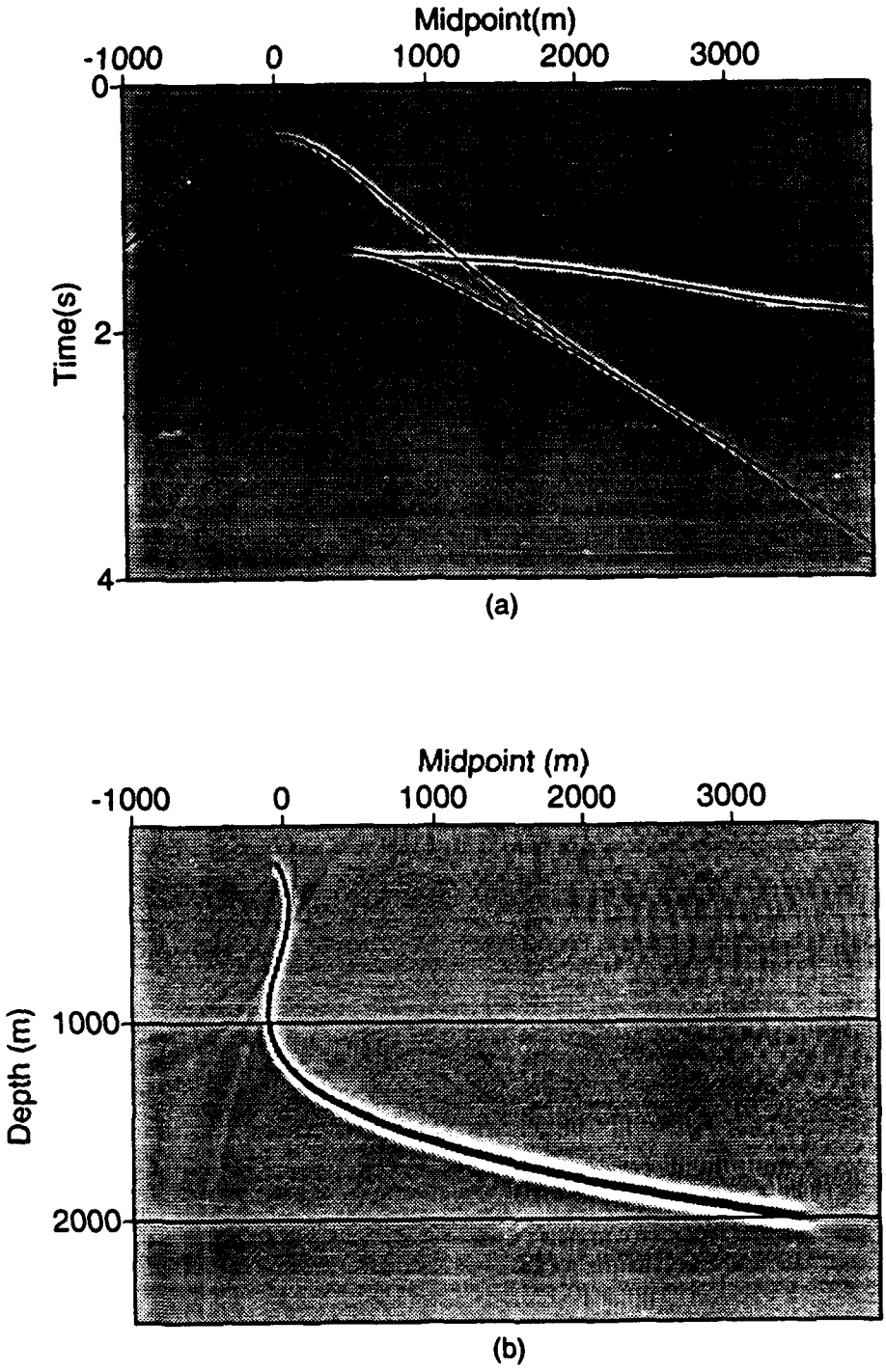

FIG. 6. (a) Data (offset=500 m) for the model in Figure 5. (b) The Kirchhoff migration of the data in (a), with velocity model in Figure 1b.

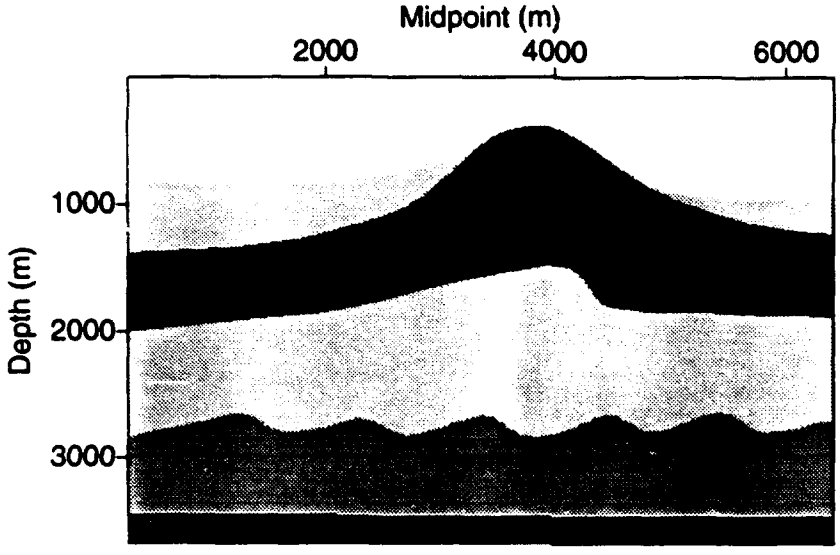

FIG. 7. True velocity model in smoothing test. The darker shading denotes higher velocity.

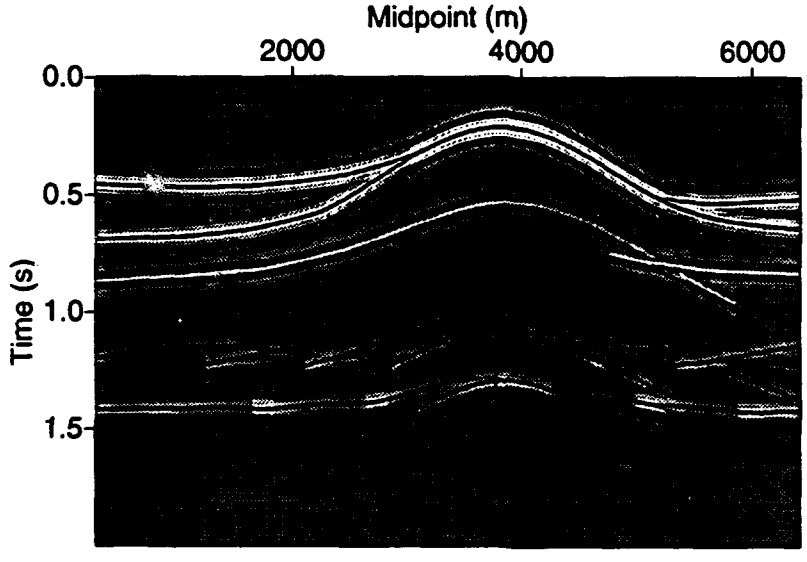

FIG. 8. Zero-offset synthetic data generated with the velocity model in Figure 7.

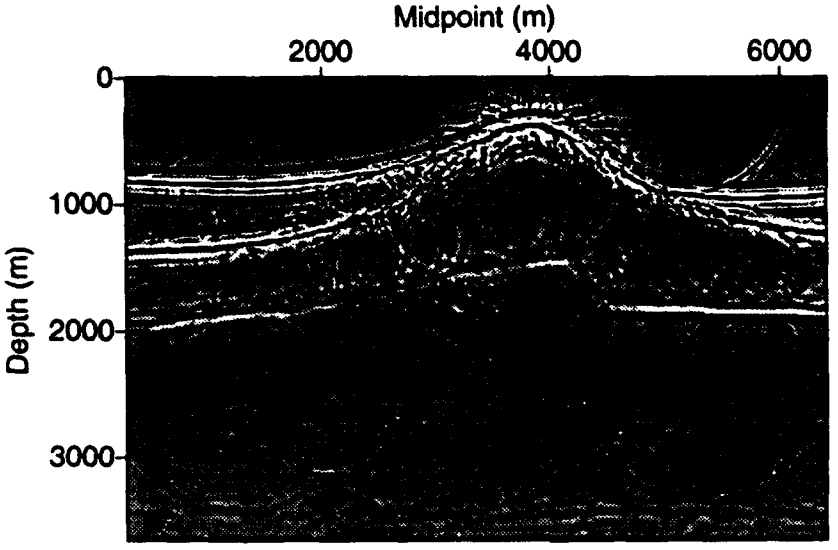

FIG. 9. Gaussian beam migration with the unsmoothed velocity model in Figure 8.

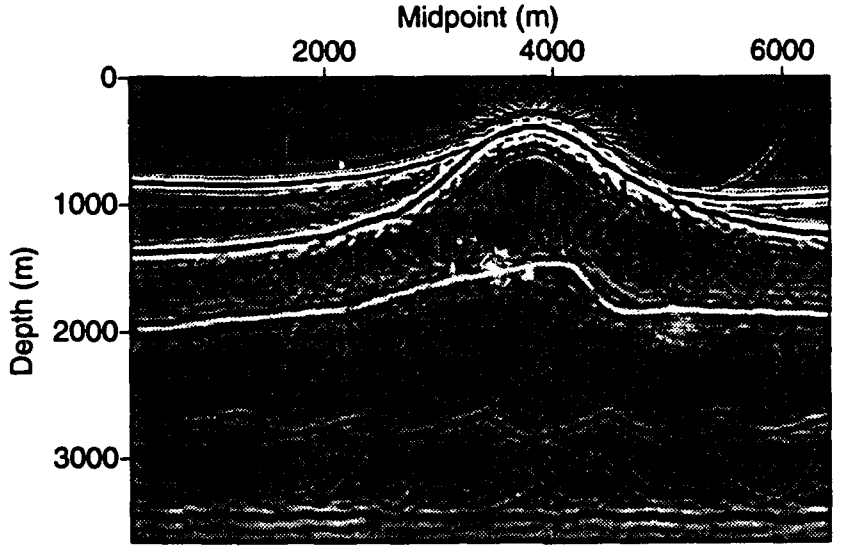

FIG. 10. Gaussian beam migration with a smoothed velocity model. The smoothing parameters $\alpha_x = \alpha_z = 2.5$. The velocity is undersmoothed.

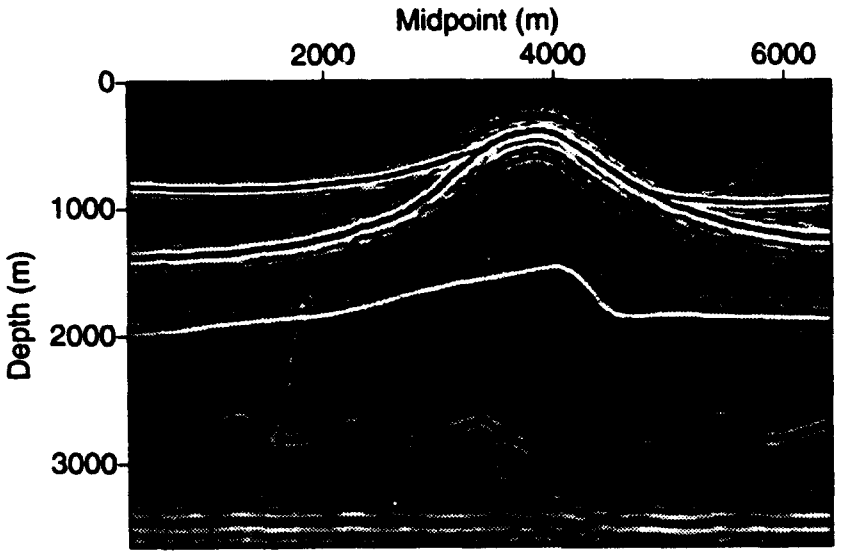

FIG. 11. Gaussian beam migration with a smoothed velocity model. The smoothing parameters $\alpha_x = \alpha_z = 5$.

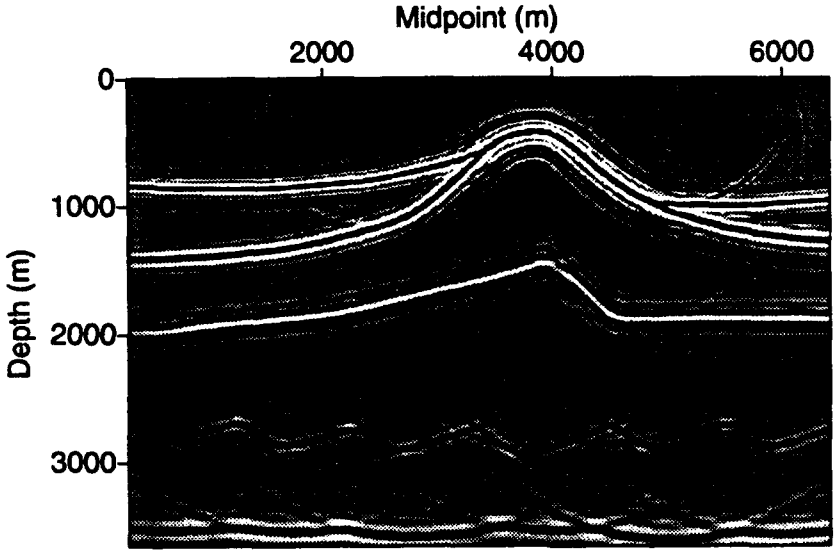

FIG. 12. Gaussian beam migration with a smoothed velocity model. The smoothing parameters $\alpha_x = \alpha_z = 20$. The velocity is oversmoothed.

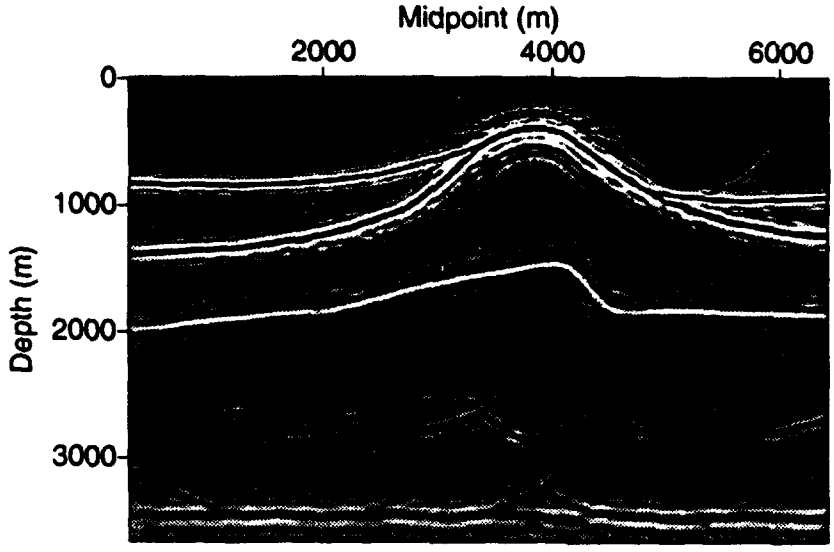

FIG. 13. Gaussian beam migration with a smoothed velocity model. The velocity is smoothed by the windowed averaging with the window size of 9 points.

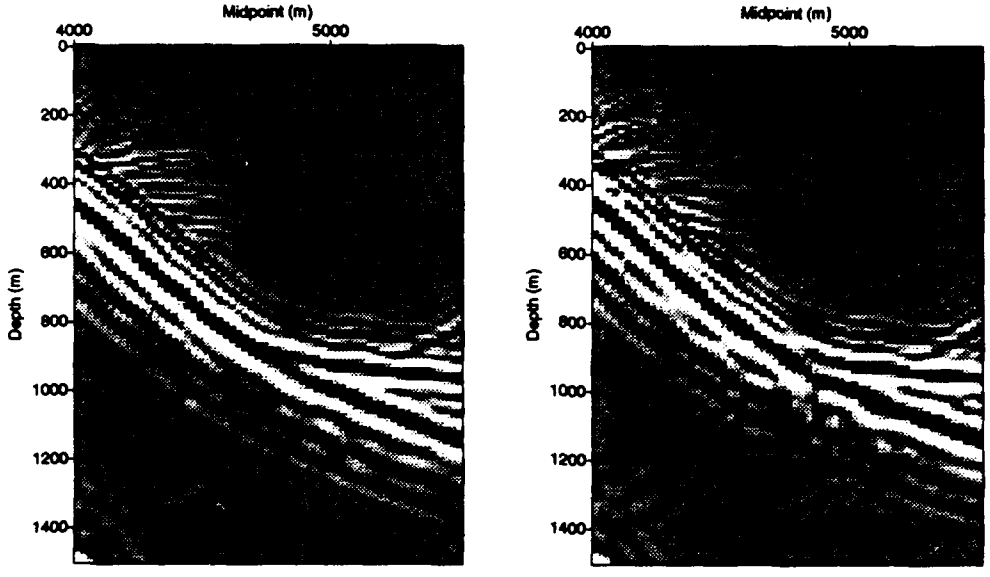

FIG. 14. Selection form Figure 11 and Figure 13. The left one is from Figure 11 and the right one is from Figure 13.

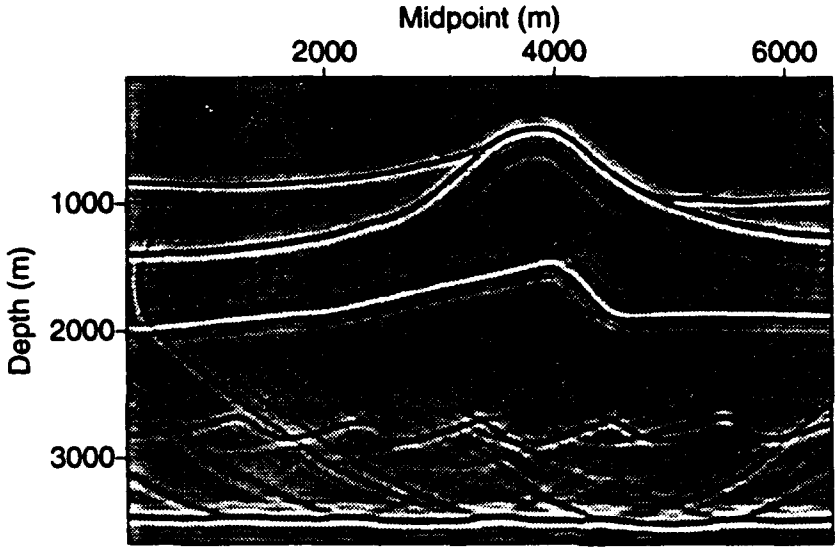

FIG. 15. Kirchhoff migration with a smoothed velocity model. The smoothing parameters $\alpha_x = \alpha_z = 14$. The velocity is slightly oversmoothed.

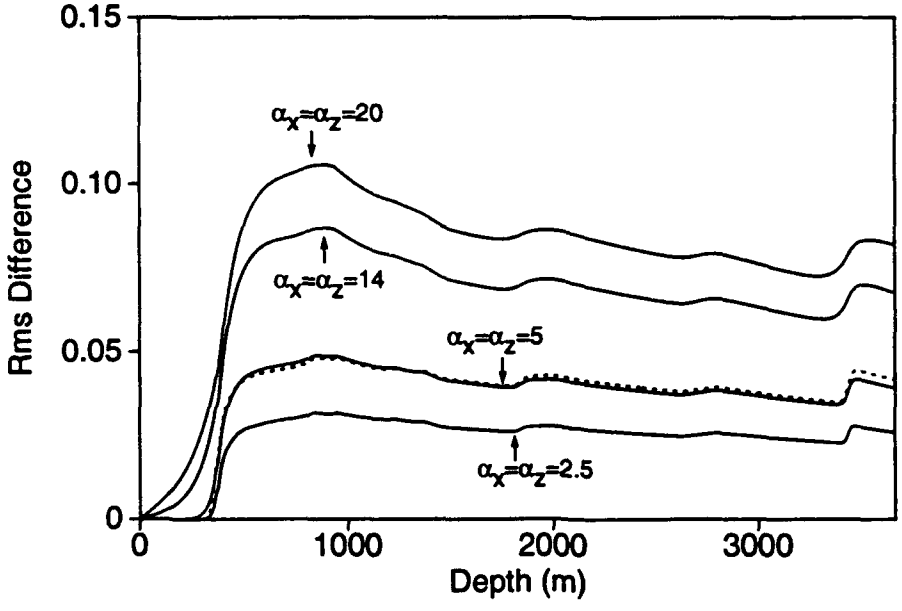

FIG. 16. The relative rms-differences between smoothed velocities and the true velocity. Four solid curves denote the smoothed velocities by the damped least-squares. The dashed curve denotes the smoothed velocity by the windowed averaging.

Taking the derivative with respect to β and using equation (A-3) gives

$$\frac{\partial x}{\partial \beta} = -\frac{v_0}{a\sin^2\beta} + \frac{v(z)}{a\sin^2\phi}\frac{d\phi}{d\beta} = -\frac{v_0}{a\sin^2\beta} + \frac{v^2(z)\cos\beta}{av_0\sin^2\phi\cos\phi} = \frac{v_0(\cos\beta - \cos\phi)}{a\sin^2\beta\cos\phi}.$$

Therefore,

$$\frac{\partial x}{\partial \beta} = \frac{\sigma}{v_0\cos\phi}; \tag{A-5}$$

i.e.,

$$\frac{\partial \beta}{\partial x} = \frac{v_0\cos\phi}{\sigma}. \tag{A-6}$$

From equation (4),

$$\frac{\partial \beta}{\partial z} = -\frac{\partial \beta}{\partial x}\frac{\partial \tau}{\partial x}\left[\frac{\partial \tau}{\partial z}\right]^{-1} = -\frac{\partial \beta}{\partial x}\tan\phi$$

The above formula and the chain rule for partial differentiation give

$$\frac{\partial \beta}{\partial \theta} = \frac{\partial \beta}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial \beta}{\partial z}\frac{\partial z}{\partial \theta} = \frac{\partial \beta}{\partial x}(r\cos\theta + r\sin\theta\tan\phi) = \frac{r\cos(\theta-\phi)}{\cos\phi}. \tag{A-7}$$

Finally, substituting equation (A-6) into (A-7), I obtain

$$\frac{\partial \beta}{\partial \theta} = \frac{rv_0}{\sigma}\cos(\theta - \phi). \tag{A-8}$$

When the velocity is a linear function of x and z, there is a rotational transformation around the point $(x_s, 0)$ such that the velocity is a linear function of z in the new coordinate system. Also it is easy to show that equation (A-8) is invariant under the rotational transformation.

APPENDIX B: SMOOTHING IN WAVENUMBER

By using the Euler's formula from the calculus of variations, we change (16) into a differential equation for $\bar{v}_s(x,z)$:

$$\bar{v}_s(x,z) - v(x,z) - \alpha_x^2\frac{\partial^2 \bar{v}_s}{\partial x^2} = 0. \tag{B-1}$$

Taking the Fourier transforms with respect to x and z gives the solution in the wavenumber domain

$$\bar{v}_s(k_x,k_z) = v(k_x,k_z)/(1+\alpha_x^2k_x^2).$$

Similarly,

$$v_s(k_x,k_z) = \bar{v}_s(k_x,k_z)/(1+\alpha_z^2k_z^2).$$

So

$$v_s(k_x, k_z) = \frac{v(k_x, k_z)}{(1 + \alpha_x^2 k_x^2)(1 + \alpha_z^2 k_z^2)}. \tag{B-2}$$

In discretization, the second derivative can be approximated by

$$\frac{\partial^2 \bar{v}_s}{\partial x^2} \approx \frac{\bar{v}_s(x + \Delta x, z) - 2\bar{v}_s(x, z) + \bar{v}_s(x - \Delta x, z)}{(\Delta x)^2}$$

whose Fourier transform with respect to x is

$$\frac{4 \sin^2(k_x \Delta x/2)}{(\Delta x)^2} \bar{v}_s(k_x, z).$$

Using this formula and taking the Fourier transforms with respect to x and z in (B-1) give the solution in the wavenumber domain

$$\bar{v}_s(k_x, k_z) = \frac{v(k_x, k_z)}{1 + 4(\alpha_x/\Delta x)^2 \sin^2(k_x \Delta x/2)}.$$

Similarly,

$$v_s(k_x, k_z) = \frac{\bar{v}_s(k_x, k_z)}{1 + 4(\alpha_z/\Delta z)^2 \sin^2(k_z \Delta z/2)}.$$

So

$$v_s(k_x, k_z) = \frac{v(k_x, k_z)}{(1 + 4(\alpha_x/\Delta x)^2 \sin^2(k_x \Delta x/2))\,(1 + 4(\alpha_z/\Delta z)^2 \sin^2(k_z \Delta z/2))}. \tag{B-3}$$